

***Application
for
United States Patent***

To all whom it may concern:

Be it known that I: Eugene M. Lee

have invented certain new and useful improvements in:

**COMPUTER-IMPLEMENTED METHOD AND SYSTEM FOR MANAGING
ATTRIBUTES OF INTELLECTUAL PROPERTY DOCUMENTS, OPTIONALLY
INCLUDING ORGANIZATION THEREOF**

of which the following is a full, clear and exact description:

**COMPUTER-IMPLEMENTED METHOD AND SYSTEM FOR MANAGING
ATTRIBUTES OF INTELLECTUAL PROPERTY DOCUMENTS, OPTIONALLY
INCLUDING ORGANIZATION THEREOF**

Inventor: Eugene M. Lee

RELATED APPLICATIONS

This application is a continuation-in-part of U.S. Application Ser. No. 10/692,793, filed October 27, 2003, which is a continuation-in-part of U.S. Application Ser. No. 10/229,273 filed August 28, 2002, which claims priority from U.S. Provisional Application Ser. No. 60/315,021, filed August 28, 2001, all of which are expressly incorporated herein by reference.

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention is directed to computer-related and/or assisted systems, methods and computer readable mediums for organizing intellectual property documents. More specifically, it relates to systems and methods for assigning attributes to intellectual property documents, managing the attributes, and filtering the intellectual property documents based on the attributes.

Description of the Related Art

Many corporations are focusing on their intellectual property assets as being quite valuable. Hence these companies strive to develop large intellectual property portfolios, and

indeed spend time and money on these assets. There is a concomitant pressure to leverage and/or better manage these portfolios of intellectual property assets. As a result, a great deal of emphasis has been placed on better ways to analyze the value of a portfolio, better processes for managing the portfolio and better strategies for creating opportunities to extract value from the portfolio.

While these management techniques have resulted in more efficient use of attorney resources, and more targeted intellectual property filings and funding, relatively little has been done to take advantage of current computational technologies, the integration of data resources (largely through the Internet), and better knowledge-based software systems to handle aspects of intellectual property. As a result, no process or product exists for handling the full range of intellectual property functions in an automated manner.

Accordingly, there exists a need in the market for a comprehensive system that incorporates tools that will give the intellectual property professional the ability to work in all aspects of their practice area using automated, analysis tools to prepare them in their practice.

Moreover, many corporations have a wide range of intellectual property assets, but no technique to make associations between assets. For example, a particular license may implicate several patents. Conventional systems do not support the association of the intellectual property assets, and they certainly do not support a memorialized explanation of the association. Even if a user is able to determine a few related intellectual property assets, the problem of determining associated intellectual property assets grows geometrically more complex with the number of intellectual property assets.

Accordingly, I have determined that the complexities affecting the analysis, use, accessing, researching, presenting, etc., of intellectual property and related information make it extremely difficult for a customer to integrate information in various scenarios. I have determined that a customer might want to determine, e.g. which patents (or other intellectual property documents) are implicated by and/or related to, for example, a license for a particular product, a particular product, a component (integrated into several products), a company, a corporate division, a particular technology, a particular sub-technology or a category under a particular technology, a type of service, or other, perhaps company –specific, criteria. I have further determined that a customer might want to ascertain the intellectual property documents that are related on multiple levels, optionally including details relating to the relationships. Further, a customer might desire to annotate one or more intellectual property documents. There exists a profound need for such a method and/or system.

BRIEF SUMMARY OF THE INVENTION

I therefore propose to provide, inter alia, methods and/or computer systems that will allow intellectual property documents and information to be associated, annotated and/or linked, including, e.g., the ability to tag (or otherwise label) intellectual property documents with attributes. Optionally, I propose that the system provide attributes and/or the user may define attributes, potentially at multiple levels. Optionally, I propose to utilize the attributes in filtering intellectual property documents, e.g., in reports. Related tools are optionally included.

Consequently, the present invention alleviates the deficiencies of conventional techniques described above. Aspects of the present invention provide for the significant improvement of the management of intellectual property, by for example, enabling personnel to label attributes of various intellectual properties and related documents of a company, its partners, and/or competitors, and/or to manage the attributes, to utilize attributes in filtering intellectual property documents, for example, in reports. An optional aspect of the present invention provides that attributes may be inherited via specifying a lower level attribute. .

By way of example, a clause within a sublicense agreement may reference a paragraph within an original license agreement, a spreadsheet containing royalty payment schedules, a copyright registration, and/or a photograph of the cover of a book. Furthermore, a clause within the sublicense may have textual and/or multimedia annotations, explanations, etc. containing comments by attorneys and/or corporate management emphasizing specific aspects of importance about the related item. Another example is a product specification referencing an individual claim of a patent, specific paragraphs of a license agreement, a registered trademark, a product web page on the Internet, and/or a link to a product drawing file. The reference to the

patent claim may contain, e.g., a conformance analysis of why the product does not violate a competitor's patent or why a particular patent may be invalid over a particular prior art reference or why a competitor's product infringes a particular patent, or a comment identifying how a key competitive feature of the product is protected by the patent.

One or more embodiments of the present invention may be capable of various configurations to adapt to user needs. For example, in one or more embodiments it may be used in conjunction with a customer site with a single server containing all data, for use by a small number of simultaneously connected users. Alternatively, one or more embodiments may be set up for use in conjunction with a multi-server distributed environment, for user, e.g., by a large number of simultaneously connected users at a large corporate use site. In another embodiment, the invention supports users connected to an Internet backbone as part of a licensed service.

One or more embodiments of the present invention may provide for readily navigating and/or annotating intellectual property documents.

One or more embodiments of the present invention optionally allow a user to download a set of related documents, for example to work offline while disconnected from the server, and to reconnect and synchronize changes with a server.

In accordance with the present invention, there are provided methods, systems and at least one computer-readable medium for managing attributes in association with electronic intellectual property documents, implemented by a computer system. At least one embodiment of the present invention includes providing one or more attribute types, having attributes including at least one attribute; providing one or more documents; associating the document(s) with the attribute(s), and determining one or more references corresponding to the document(s);

associating one or more other documents with the attribute(s) and determining one or more other references corresponding to the other document(s); and storing the reference(s) and the other reference(s) in association with the attribute(s), for later retrieval of the document(s) and/or the other document(s).

Optionally, one or more embodiments of the present invention include determining one or more groups of documents responsive to a user, the group including the document(s) and/or the other document(s).

Optionally, one or more embodiments of the present invention include organizing the group(s) of documents, wherein the documents are organized by one or more of: at least one field therein, and at least one attribute previously associated therewith. Optionally, the group of documents omits a lapsed document.

One or more embodiments of the present invention includes assigning the document(s) to one or more other groups. According to one or more embodiments of the present invention, the document(s) is: XML format, binary format, image data, audio data, an interpretive file, and/or video data.

Optionally, one or more embodiments of the present invention includes searching for the document(s) based on criteria including the attribute(s) and/or the attribute type(s).

One or more embodiments of the present invention provide for accessing the document(s) from the attribute(s) and the reference(s) stored in association therewith. One or more embodiments of the present invention provide for displaying information characterizing the document(s).

Optionally, the attribute(s) is associated with one or more users, the document(s) being accessible by users including the mentioned user, further comprising limiting access to the attribute(s) to the user(s) associated therewith. Optionally, the attribute(s) further includes: a reference to a URL, a reference to another file, and/or user-provided text.

According to one or more embodiments of the present invention, the document(s) and the other document(s) are representative of: an invention disclosure document, a patent document, a trademark document, a copyright document, a product description document, a contract document, a license document, a sui generis protection document, a design registration document, a trade secret document, and/or an opinion document.

One or more embodiments of the present invention provides for preliminarily determining (or customizing), for the attribute type(s), the attribute(s).

One or more embodiments of the present invention provide for storing the document(s) and the other document(s) for later retrieval.

According to one or more embodiments of the present invention, the attribute(s) is selected from attribute types representative of: a product and/or a service. Optionally, the attribute types further include: an actor, a user entity, a current owner, and/or a project.

One or more embodiments of the present invention provides one or more sub-attribute type associated with the attribute type(s), and wherein the sub-attribute type(s) has second attributes including one (or more) associated with the attribute(s), wherein the attribute type(s) inherits the attribute(s), responsive to an association of the second attribute(s) with the document(s).

Optionally, one or more embodiments of the present invention provide for exporting the attribute(s) and the attribute type(s).

One or more embodiments of the present invention provide for utilizing the attribute(s) as criteria for: searching, retrieving, reporting, and/or viewing the document(s).

According to one or more embodiments of the present invention, the attribute(s) is utilized in combination with one or more of: (i) the attribute type(s), (ii) one or more sub-types of the attribute type(s), (iii) a content of one or more fields in the document(s); (iv) a type of one or more fields in the document(s); and/or (vi) information derived from the field(s) in the document(s).

Optionally, the reference(s) corresponds to the serial number of the document(s).

There has thus been outlined, rather broadly, the more important features of the invention in order that the detailed description thereof that follows may be better understood, and in order that the present contribution to the art may be better appreciated. There are, of course, additional features of the invention that will be described hereinafter and which will form the subject matter of the claims appended hereto.

In this respect, before explaining at least one embodiment of the invention in detail, it is to be understood that the invention is not limited in its application to the details of construction and to the arrangements of the components set forth in the following description or illustrated in the drawings. The invention is capable of other embodiments and of being practiced and carried out in various ways. Also, it is to be understood that the phraseology and terminology employed herein are for the purpose of description and should not be regarded as limiting.

As such, those skilled in the art will appreciate that the conception, upon which this disclosure is based, may readily be utilized as a basis for the designing of other structures, methods and systems for carrying out the several purposes of the present invention. It is important, therefore, that the claims be regarded as including such equivalent constructions insofar as they do not depart from the spirit and scope of the present invention.

Further, the purpose of the forgoing abstract is to enable the U.S. Patent and Trademark Office and the public generally, and especially the scientists, engineers and practitioners in the art who are not familiar with patent or legal terms or phraseology, to determine quickly from a cursory inspection the nature and essence of the technical disclosure of the application. The abstract is neither intended to define the invention of the application, which is measured by the claims, nor is it intended to be limiting as to the scope of the invention in any way. These together with other objects of the invention, along with the various features of novelty that characterize the invention, are pointed out with particularity in the claims annexed to and forming a part of this disclosure. For a better understanding of the invention, its operating advantages and the specific objects attained by its uses, reference should be had to the accompanying drawings and descriptive matter in which there is illustrated preferred embodiments of the invention.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING(S)

The above-mentioned and other features and advantages of the present invention will be better understood from the following detailed description of the invention with reference to the accompanying drawings, in which:

Figure 1 is a functional block diagram illustrating a system architecture providing for annotating intellectual property documents and data, according to one or more embodiments of the present invention.

Figure 2 is a block diagram of one example of a system for use in connection with analyzing and managing intellectual property documents, according to one or more embodiments of the present invention.

Figure 3 is a combined functional block diagram with data flow, illustrating an exemplary data manager according to one or more embodiments of the present invention.

Figure 4 is a user interface, illustrating editing of an intellectual property document, according to one or more embodiments of the present invention.

Figure 5 is a user interface illustrating an example of annotation for an intellectual property document, according to one or more embodiments of the present invention.

Figure 6 is a user interface illustrating an example of an additional window for linking a selected document to another intellectual property document, according to one or more embodiments of the present invention.

Figure 7 is a user interface illustrating an example of a report view of a marked-up document, according to one or more embodiments of the present invention.

Figure 8 is a user interface illustrating an example of a map view of a marked up document, according to one or more embodiments of the present invention.

Figure 9 is a functional block diagram with data flow, illustrating an exemplary data server according to one or more embodiments of the present invention.

Figure 10 is a functional block diagram with data flow, illustrating an exemplary data analyzer according to one or more embodiments of the present invention.

Figure 11 is a block diagram, illustrating data flow for splitting an annotated document into annotated data and document data, according to one or more embodiments of the present invention.

Figure 12 is a block diagram illustrating data flow for merging the annotation data and document data of Figure 11 into the annotated document, according to one or more embodiments of the present invention.

Figure 13 is a block diagram illustrating an example association of external data with a document that has been annotated, in accordance with one or more embodiments of the present invention.

Figures 14A-B are a flow chart illustrating merging of document data with annotation data to produce a marked-up representation of the document, according to one or more embodiments of the present invention.

Figure 15 is a flow chart illustrating splitting a marked-up representation of a document into annotation data and document data, according to one or more embodiments of the present invention.

Figure 16 is a linked diagram illustrating an example of annotated intellectual property documents and data, according to one or more embodiments of the present invention.

Figure 17 is a linked diagram illustrating another example of annotated intellectual property documents and data, according to one or more embodiments of the present invention.

Figures 18A-B are a flow chart of an example of annotating a document and/or linking the document to another document (or portion thereof), according to one or more embodiments of the present invention.

Figure 19 is a flow chart illustrating an example of traversing intellectual property documents via links in an annotation thereto, according to one or more embodiments of the present invention.

Figure 20 is a block diagram illustrating a computer architecture, for use in connection with one or more embodiments of the present invention.

Figure 21 is an illustration of a computer appropriate for use in connection with one or more embodiments of the present invention.

Figure 22 is a block diagram illustrating the internal hardware of the computer of Figure 21.

Figure 23 is an illustration of an alternative computer appropriate for use in connection with one or more embodiments of the present invention.

Figure 24 is an example flowchart for determining attributes to be associated with intellectual property documents, according to one or more embodiments of the present invention.

Figure 25 is an example flowchart for grouping intellectual property documents into a project, according to one or more embodiments of the present invention.

Figure 26 is an example flow chart for using attributes to filter through intellectual property documents, according to one or more embodiments of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The following detailed description includes many specific details. The inclusion of such details is for the purpose of illustration only and should not be understood to limit the invention. Throughout this discussion, similar elements are referred to by similar numbers in the various figures, for ease of reference. In addition, features in one embodiment may be combined with features in other embodiments of the invention.

Intellectual property information regarding intellectual property documents by a customer, intellectual property service provider, government entity or other source has been collected. Likely such information is extracted and deposited into one or more databases. Ultimately at least a portion of such intellectual property information is presented to an end user on behalf of a customer, such as via an intellectual property application, which may be executing locally, or via a web site over the World Wide Web, i.e., the Internet. For ease of description, such a collection of information will be referred to herein as “database”, although it should be

recognized that the information might be collected in other formats as well, and that an intellectual property application might not be restricted to data stored in a database.

The association between selected intellectual property information is realized and optionally annotated. The annotation enables users to annotate images and text in intellectual property, such as, e.g., patent drawings. Those annotations are saved and optionally categorized. For example, annotated drawings or images are saved in the context of projects in order that notes and other thoughts of the user are memorialized and tied to a project.

Figure 1 is a functional block diagram illustrating a system architecture providing for annotating intellectual property documents and data, according to one or more embodiments of the present invention. In the illustrated example, the system is realized as an intellectual property portal 111 on a general purpose computer, communicating with a network, e.g., the Internet 105. A user 107 accesses the portal 111 via the Internet 105. A document workspace 109 is provided on a computer for the user 107. Applications for locating, viewing and annotating intellectual property documents and data are provided on the portal 111. In the present example, the applications include an editing view 119, an annotation view 117, a map view 115, a report view 113, a search application 101, and a browse application 103. Any application program useful for searching or browsing intellectual property documents may be utilized to implement the search application 101 or the browse application 103. Intellectual property documents and data may be stored in any appropriate manner. In the present example, the intellectual property documents and data are stored in a patents database 131, a trademarks database 125, a copyrights database 127, a licenses database 129, and an opinions database 123. In this example, the opinions database 123 is separate from the portal 111. The system also includes storage of the annotations in an annotations database 121. The user accesses one or

more of the intellectual property documents together with any annotations, e.g., by searching or browsing for the selected document. The user may manipulate, annotate, and/or link one or more selected documents via one or more of the applications. The annotated and/or linked document(s) are stored into the appropriate databases by the user. Access to annotations and/or annotated documents optionally is limited, e.g., by corporate affiliation of the user and annotation, by user, by express permission to one or more users, etc.

A system design for use in connection with one or more embodiments of the present invention, as illustrated in Fig. 2, may advantageously comprise a number of interconnected components. Each component may focus on a specific task, and advantageously provides and/or utilizes an Application Programming Interface (API) to communicate with other components within the system, according to the illustrated realization of one example system. Components may include, for example, one or more data servers 205-211, a data manager 201, one or more analyzer views 113, 115, 117, 119, a management console 221, and/or a watch agent 223. Components and/or functions thereof may be omitted, replaced, subdivided and/or combined and still remain within the scope of the invention. The data servers 205-211 provide access to the data representing the intellectual property documents; the data analyzer 203 provides user interfaces to obtain, analyze and/or traverse intellectual property documents; and the data manager 201 breaks down documents into storable units and builds up documents for the user interfaces.

In accordance with one or more embodiments of the present invention, scalability may be provided by the logical and/or physical separation of data server 211 functionality from the data manager 201. For example, the data manager may reside with one (or more) data server on e.g., a single machine storing all patent, licensing, and annotation data. As one of many alternatives,

the data manager may connect to multiple data servers, each running on a separate machine, and each storing only a portion of the data.

According to one or more optional realizations of the present invention, offline storage and/or operation may be provided for example, by a document manager, discussed below, which stores some or all working data locally on the user's machine, and/or by API functionality to retrieve and store documents from the data manager 201.

Figure 2 illustrates one or more embodiments of a general overall architecture for use in connection with the present invention. This figure illustrates internal architecture, useful for illustrating the concepts in relation to the invention. Portions of the architecture may be omitted and/or replaced and/or combined when used in connection with certain embodiments of the present invention. This example of one or more embodiments of the present invention illustrates an optional 3-tier architecture, including the data server tier, the data manager tier, and the data analyzer tier. More or fewer tiers may be utilized, in other embodiments of the present invention. The data server (or multiple data servers) 205-211, according to one or more embodiments of the present invention, may provide for storage, versioning, indexing and/or searching of (possibly a subset of) document data (e.g., XML) annotation data, and/or image data.

Optionally, there may be provided one or more data servers 205a-c, 207, 209, 211 amongst which the data server functions may be divided. In the present example, several data servers are provided. Optionally, a multiple data server format may house one or more sets of related information. In this illustration, one data server might contain the entire United States Patent and Trademark Office ("USPTO") database; and/or one data server might contain

multiple databases. According to one example alternative, the related information (in this case, the USPTO database) may be spread over multiple data servers 105a-c. According to a convenient realization, for example, one data server may contain only the patents ending in “1”, another server might contain patents ending in “2”, a third might contain patents ending in “3”, and so on; according to this example, there are 10 servers, across which is distributed, preferably in a logical manner, preferably the entire USPTO database. By distributing the data servers and functions, one or more embodiments of the present invention may provide for a scalable solution for storing generalized data used by the system.

According to one or more embodiments of the present invention, the data may be stored in its original format. Alternatively, it may be reformatted at some point or points prior to storage. The format for the data that the USPTO currently provides data for patents is XML, a mark-up language which is fairly similar to HTML. XML is a generalized syntax for creating a document structure and tags, unlike HTML, which has predefined tags. XML essentially leaves the meaning of those tags to the developer of the dialect. In this case the USPTO has defined the individual tags that exist within this language and the meaning tag of each. The system may use the syntax as provided by one or more patent/trademark offices, government, and/or commercial data providers, or optionally, the syntax may be converted, e.g., into one or more standard formats. In the illustrated example, the USPTO patent database (both text (XML) and image data) is distributed across three data servers 205 a, b and c.

Similarly, one or more embodiments of the present invention may accommodate other data and/or other formats, e.g., an XML schema for license agreements. Such a schema for a license agreement may accommodate, e.g., typical, usual, optional and/or advanced elements that are available within the license agreement, e.g., a preamble, definition section, individual

definitions, paragraphs, clauses, sections, articles, etc. In the illustrated example, license documents are stored on the data server 209.

As illustrated in the example of Figure 2, each data server in a multi-data server embodiment within the system may contain all, a subset and/or a portion of the information that is available to the user. Data server 5 209 stores, in the present example, license data, copyright data, and trademark data. These databases are likely to be much smaller than the USPTO patent database. Hence, a single server may store more than one type of data. Optionally, non-USPTO data is included.

According to the illustrated example, data server 4 207 stores annotation data (discussed below), e.g., having annotations corresponding to some of the patents and/or licenses. The annotation data may include, e.g. electronic mark-ups that attorneys or other users would make, e.g., in connection with a document. Further in this example, data servers 1 through 3 store the patent text and image data of the USPTO patent database (or a portion thereof).

The optional data manager 201 may pull together the data that may be distributed across one or more servers. The data manager advantageously provides a single cohesive and comprehensive management of a given database. The data manager, according to one or more embodiments of the present invention, provides for the seamless distribution, coordination, and searching, of documents (e.g., XML), or merging of annotation data (e.g., XML), and image data across one or more data servers. It optionally may support caching of search requests and/or results, and/or replication of data to and/or from remote servers.

Reference is made to Figure 3, providing one or more example embodiments of the data manager 201 architecture. The data manager may provide an object API 319 having services to

receive requests and/transmit information to/from the data analyzer 323, e.g., to insert, update, delete, and/or request document data, e.g. XML data. The object API may have and/or retrieve binary data, such as for images and/or sound, for example. XML data requests may be further processed within the data manager; and, if appropriate, passed to a data server 321; binary requests may be passed on through to the underlying data server 321.

Consider for example that documents are provided in XML format, and that annotations for each document are provided as annotation data entities therein. When an XML data insertion or update occurs, the XML data is first parsed by an XML parser 308. This parser maybe driven by a mark-up schema 317, which identifies XML tags within the document for annotation data entities, and the relationship of the XML tags to the document data entities. The annotation data entities are extracted from the XML document. They may be used to create an annotation XML data stream. The remainder of the XML data, that is, the potentially revised document without the annotations, may be used to create a document XML data stream.

Where multiple data servers are provided, an optional connection manager 301 may be provided, to identify which data server(s) stores the data at issue if distributed, e.g., the document data, the annotation data and/or the image data, such as by maintaining a mapping. Image data may be stored on the same data server as the document data or may be stored elsewhere.

Continuing with the above example of XML documents, when an XML request occurs, the data manager 201 retrieves the document XML data and the annotation XML data from their respective one or more data servers. It then parses both these XML data streams with one or more XML parsers 308. Using the mark-up schema 317, it embeds the annotations from the

annotation XML data within the corresponding tagged elements of the document XML data, and with annotation merge logic 307, merges both streams into a single XML document.

When a search request occurs, search and result merge logic 303 optionally looks up each keyword in the one or more thesauri of the data server(s), and any match is added to a search keyword list. The search request may contain a list of searchable fields appropriate to the documents being searched (e.g. Abstract, Inventor, Claims, etc. for patents), and/or the scope of the search (e.g. Patent, Copyright, Annotation, etc.). The search is then executed on the relevant Data Servers, the results are collected, and they are returned to the caller. Search results optionally may be returned from the data server in partial result groupings, such as of a specified fixed size; this permits the data manager 201 to satisfy a search request quickly, while deferring much of the processing overhead for result fetching until actually needed.

If a user is browsing back and forth through a number of items returned from a search, it is likely that they will request the same document repeatedly within a short period of time. An optional cache manager 311 maintains a mapping of client search requests to search results. If a request is repeated while the result is cached, the result may be returned from an Image and XML cache 313 through the cache manager 311, instead of generating a new data server search request.

When the optional change notification event is received from a data server, it is passed in to the change notification handler 309, then through to the object API 319. It may be passed to an optional replication service, which maintains a list of registered downstream data managers. One or more data managers may be registered for replication of information, as identified e.g., by data server and item type, and will be notified of such changes. A notified data server may

request the information. The replication service 315 maintains a queuing 316 of notifications for those registered data managers that are unreachable or are flagged for queuing. The optional change notification event may provide the basis for a subscription service, in order to provide customers with updated latest patent and trademark information. The optional change notification event and replication service may be used for enabling a system to distribute to multiple data servers, even if distributed around the world, while maintaining synchronization between them. The optional change notification event may be provided to the cache manager 311, which may be used to enable it to flush an image and XML cache 313 of outdated items.

Reference is made back to Figure 2. According to one or more embodiments of the present invention, an optional search engine locates stored documents by performing searches on phrases and/or individual words. For example, the search engine interface may provide a column for proceeding word and another for following word. As a further example, to access intellectual property patent data, when doing a search, a search request may result in a hit to all three of the Data Servers 205 a, b, c in parallel. According to one or more embodiments of the present invention, the data manager 201 is responsible for coordinating among the distributed Data Servers where multiple data servers have potentially relevant data, and for being aware of the range of specific data on each data server. If a search request is received, the data manager 201 may broadcast that request to all of the relevant data servers (three in the illustrated example), receive the search request results returned from those data servers, and then merge the results back together again and create a single common results set. The advantage of distributing a search is that one may speed up the search, average-out the effect of multiple users, and/or numerous requests being received, and load-level the users working with an individual patent and its image data. The data manager 201 thus provides an optional second logical level, where

it pulls together the content of the data servers, and/or provides among other things a view into a company's intellectual property database.

An optional third logical level is the data analyzer 203. The data analyzer 203 performs, inter alia, formatting of information into a representation that is user friendly, so that a user may read and/or edit. The data analyzer 203 may include prompting the user for annotations, for accepting annotation data, for displaying data, for creating reports, for creating a document map which demonstrates the relationships of one set of information to another, etc.

Reference is made to Figures 4-8, illustrating several example windows 401, 501, 601, 701, and 801, open within a user interface according to one or more embodiments of the present invention. One or more aspects of the present invention assist in working with relationships between documents and/or portions thereof.

The user interfaces to the intellectual property documents are optionally enabled by the data analyzer 203, illustrated in Figure 2. Reference is made to Figures 1 and 4-7. In this example, the license in the editing window 401 (Fig. 4) correlates to the editing view 119 (Fig. 1). A report window 701 (Fig. 7) demonstrates the report view 113 (Fig. 1), a map window 801 (Fig. 8) demonstrates the map view 115 (Fig. 1) and a mark-up window 501 (Fig. 5) represents the annotation view 117 (Fig. 1).

Reference is now made to Figure 4. The user in this example has retrieved an intellectual property document to edit and/or annotate, e.g., a license, into an editing window 401. One or more aspects of the present invention provide that the user may logically subdivide that document into sections. Those sections may then be related to sections within that or another document. The relationship between one document and another, and/or between one section in

one document and a section in another document (or the document itself) may be annotated. The annotation allows a user, e.g., an attorney who is analyzing this information, to indicate within a document, for example, an issue, the result of an analysis, how this portion of this document relates to that portion of that document, etc.

Referring again to Figure 4, the intellectual property document 403 (in this example, the license) is displayed in the editing window 401. The editing window 403 presently displays that portion of the document encompassing “Article 1,” “Section 1.1,” which in the example is entitled “Trade Secret License.” In one or more embodiments of the present invention, one or more active portions 405, e.g., “Trade Secret License”, may be outlined, and/or highlighted such as in red on the screen, in order to indicate that this is an active portion 405 of the document being viewed. A further indication, e.g., a special highlight or color, e.g., optionally may be used to indicate that there is an annotation associated therewith, e.g. a possible conformance issue or a failed conflict.

By way of example of a possible use of one aspect of the invention, if a user is performing, e.g. evaluation of a license against a patent or a product against a patent, for each of the claims in the patent, the user may be viewing parts of the license and the claims one at a time and indicating that a certain aspect of this product, license, or document fails to conform to some aspect of this patent claim. The user may select one of several standard notations reflecting, for example, a standard, system provided relation, and/or a super-user-customized attribute concerning the respective documents, e.g. that a product or license, etc. is in violation of this patent claim, or may be in violation of this patent claim, or is not in violation of this patent claim. The user may wish to add other text, annotations, references to other documents or URL's or files, etc. to the document being viewed. Those thoughts, however they may be phrased or

indicated, are important to capture. An attorney or other user going through an intellectual property document, such as a patent, may indicate that a product, license, etc. does not violate this patent, claims 1, 2, 3, etc. because of annotated reasons, or indicate the need to look into this further, and/or indicate the need for a second opinion or any other indication as desired. Multiple users may each provide separate annotations.

The attorney or other user may review, edit and/or annotate an agreement or other intellectual property document in the editing window 401 for example by selecting a section, or traverse the document section by section. (The document may be subdivided previously, currently, and/or subsequently into sections automatically (e.g., within the XML format) manually, and/or semi-manually.) In the present example, beginning, e.g., with Section 1.1 the user may select a portion of the document in the editing window 401 to add an annotation or mark-up data.

Reference is now made to Figure 5, showing an example mark-up window 501, to interact with the user to obtain annotations. The mark-up window 501 pops up in response to a user indication that he wishes to annotate a document (or portion thereof). In the present example, the user may select one or more type of pre-defined notations, e.g., “conformance” 503, view “notes” 507, view a history 509 of changes to this section, and/or view some user-defined attributes 511, and/or categories or links to images or web pages, etc.

In the illustrated example, “Harvey Wallenbarger” is the user and selects Section 1.1 in the editing window 401 (shown in Figure 4). In response to the selection, the system obtains the user’s annotation via the mark-up window 501. In the mark-up window 501, in “conflicts” under the “conformance” tab 503, the user selects “possible” indicating that there is a possible

conformance violation; the user may alternatively or in addition type in text comments, e.g., to memorialize concerns about the possible conformance violation. By selecting at the top of the mark-up window 501, one or more embodiments of the present invention includes a drop down list box or chooser 505 that provides a mechanism for choosing a related intellectual property document, for example one of several documents that the user may be working with, thereby relating the section and/or its annotation to a section of another (or the same) intellectual property document. In the illustrated example, the user notes a relationship between the annotated license section 1.1 to a section of another intellectual property document.

Optionally, the other document or other section of the same document is displayed in an optional further editing window 601, shown in Figure 6. Optionally, a selected section 603 of the related document is highlighted.

According to one or more embodiments of the present invention, annotation is realized as a manually-driven approach. For example, a user goes through a document one item at a time and performs an annotation. The process of annotating is preferably a manually-driven process, for several reasons. For example, one person may use the term “cup” but another person may choose to use the designation “a liquid containing dispensing container” for the same object. To create an automated mapping between those two designations may be possible, using for example a thesaurus, where the user may add synonyms that expand the scope of the search, etc. Nevertheless, to be able to parse-out the complex language that tends to appear in intellectual property documents, and to be able to accurately perform an analysis against similarly complex wording by a completely different person is, may be better done manually or semi-automatically.

Reference is now made to Figure 7. One or more embodiments of the present invention optionally provide for a report window 701. The report window 701 provides a summary of the mark-ups to, e.g., the selected document. In this example, the report window 701 includes a summary 709 of mark-ups including a count of sections and types of mark-ups. Optionally, each section 703 and sub-section 705 also is summarized. A section or subsection summary optionally includes a mark-up summary 707, with, e.g., the standard notation type, any reference(s), author, date, and/or other annotation data. The present example indicates that one or more users has reviewed this license (or other document), checked it against a particular document or documents, and summarized some or all of the mark-up data and associated portions of the document that have been annotated.

The optional map window 801 illustrated in Figure 8 provides a map of the sections of the mark-up document 809 and the related intellectual property documents noted in annotations. In the present example, the map window 801 includes a visual representation 803 of each document section and a summary 805 of each related document noted in a mark-up. A map line 807 indicates a relation between document sections and subsections, and a connection 808 is indicated between documents and sections/subsections.

Other components, plug-ins, reports, and/or tools may be provided to view, search, edit, annotate, link and/or mark-up intellectual property documents, in accordance with one or more embodiments of the present invention. The view and window functionality, for example, may be combined, omitted, and/or replaced and/or implemented in an alternative user interface in alternative embodiments of the present invention.

Reference is again made to Figure 1. The mark-up data optionally is stored logically and/or electronically embedded within the original document, e.g., as an annotation. According to one or more embodiments of the present invention, the system may host most or all of the databases 121, 125, 127, 129, 131 on web servers, for use by any of a large number of users. According to one or more embodiment of the present invention annotated documents are stored, e.g., locally and/or remotely. According to one or more embodiments of the present invention, intellectual property documents are linked to annotations, and the documents are shared.

If the users are unaffiliated with each other, however it would be undesirable to have these unrelated users accessing, e.g., the same patent, marking it up, and physically embedding additional information therein. It would not be suitable to make that information available to all of those users. Consequently, according to one or more embodiments of the present invention, the mark-up data is maintained so that the separate and/or unrelated users are protected from disclosure to each other. Accordingly, the mark-up data optionally is separately maintained from the document data and is correlated to a user and/or group of users.

Further, the mark-up data preferably is seamlessly associated with the document information, and according to one or more embodiments of the present invention is preferably presented to the user as a unitary document. Despite the unitary appearance, when the user is finished working on this document, the document and mark-up information optionally is broken into components, optionally each being stored in the appropriate and/or separate storage. Optionally, the document and mark-ups are stored together.

According to one or more embodiments of the present invention, there are provided two (or more) streams of data, corresponding respectively to the mark-up(s) and the document(s).

These data streams are merged together into a single document, and that merged document is presented to the user and/or worked on as a single logical document. When that work is complete or the user otherwise is done, then the document is split up into two (or more) different streams corresponding to the mark-up(s) and the document(s). Preferably, the document is in XML format, but could be in other formats.

Reference is made again to Figure 3, a block diagram for one or more embodiments of a data manager 201, also showing communications to/from a data server 321 and to/from a data analyzer 323. The data manager 201 may, in accordance with one or more embodiments of the present invention, provide for splitting and merging annotations and documents. In the illustration of Figure 3, a document with annotations is split into data streams via annotation split 305, and merged into an annotated document via annotation merge 307. Consider an example data flow through the data manager 201: an “XML request” from the data analyzer communications 323 is received by an object API 319. The “XML request” indicates a particular XML intellectual property document (optionally with annotations) to be retrieved, e.g., to be accessed by the data analyzer. The request is received by annotation merge component 307 in the data manager 201. The data manager 201 determines that it needs to obtain one (or more) XML document corresponding to the document data for the intellectual property document, and also one (or more) XML document corresponding to the annotation data. The annotation merge component 307 issues a request to retrieve these two (or more) documents. Consider that one of these, for illustration purposes, is a patent document and the other is annotation data marking up the patent. The annotation data includes, within the set of its information, an association between one or more individual annotations, and the location of the item or section within the patent document (the “entity”) that the annotation refers to, for example, specific claims in a

patent. So, if (as in this example) the user has annotated a particular claim in the specified patent, then the annotation includes a reference corresponding to the identifier for the entity corresponding to that claim. (There are a number of ways by which an “entity” within a document could be uniquely identified, e.g., offset from document start, logical division, etc.) According to one or more embodiments of the present invention, the annotation merge component 307 processes document data and annotation data (e.g., with an XML parser 308), identifies the one or more entities, within the document with a particular annotation, extracts the annotation (e.g., as an XML mark-up fragment), and embeds the annotation within the section of the document (e.g., an XML section) for the referenced entity within the document.

In accordance with one or more embodiments of the present invention, there are provided two (or more) different documents, one containing annotations and the other containing the document, both including a respective series of entities. The annotation document(s) is broken up into the individual entities; the documents are parsed and it is determined where the annotation entities go in the document; and the document is fattened into a marked-up document. The fattened mark-up document is then returned to the data analyzer as the document in the proper format (e.g., XML) via data analyzer communication 323.

The data analyzer then may, at that point, work with the mark-up document as if it is a single document. That the marked-up document originated from two or three or more different sources, according to one or more embodiments of the present invention, is transparent to the data analyzer. According to one or more embodiments of the present invention, the data analyzer receives, processes, and/or acts on the marked-up document as a unitary document, and when done, returns it as a unitary document. Optionally, the data analyzer works with the

document encompassing more than one file, e.g., separate document and annotation data, multiple files for document sections, etc.

According to one or more embodiments of the present invention, the data manager 201 includes one or more annotation split components 305, optionally driven by a mark-up schema 317. The mark-up schema 317 identifies which types of entities belong in a document (e.g., a patent) and which types of entities belong in an annotation. In scanning through the mark-up or expanded document, the system may identify the one or more entities that are an annotation entity. The schema identifies the annotation entities, such as in XML. Further, the system can identify that a particular annotation entity is related to a particular parent document (or entity within a parent document) and may obtain the unique identifier for that parent associated back with the annotation entity. It may then start building a new annotation document. So, in this way the system then supports the collapsing of the expanded mark-up document from the analyzer back into its normal form, extracting the annotations, building another annotation document, and then inserting data for the annotation and/or document back into the Data Server.

In the case of a patent, for example, the original document may be marked as read only, so the user cannot edit the original document. Optionally, the annotation split logic determines whether the document is read-only, thereby avoiding the need to examine edits to the original document, e.g. the original patent document. Consequently, for a read-only document, the annotation split logic 305 may review the mark-up document to extract the annotation information.

Reference is now made to Figure 9, illustrating an example block diagram of one potential embodiment of the data server 205. According to one or more embodiments of the

present invention, one or more data servers 105 retrieve/store documents and/or annotation data. According to one or more embodiments of the present invention, the intellectual property documents are stored separately from the annotation data. If desired, stored documents may be further subdivided, e.g., by intellectual property type (e.g., patent, license, trademark), or file format (e.g., XML, .TIFF, .DOC)

Each data server may advantageously provide an object API 319 which, inter alia receives communications to/from a data manager 927, to insert, update, delete, and/or request data in a format appropriate to the document(s), annotation and/or image data, e.g., XML. Hence, where documents and annotations are stored as, e.g., XML formatted data streams, the data server may act as a repository for document and/or annotation data.

When an XML data insertion or update occurs, the XML data advantageously may be stored within an XML repository 905, e.g., as a new revision of the document. The data server 205 receives the document or documents for storage, e.g., through an XML update request. An XML update request is received through the object API 319 and is optionally sent to a data versioning manager 917 to handle version updates.

According to one or more embodiments of the present invention, revisions may be managed by a data versioning manager 917. For example, when a data request occurs, the specified data may be retrieved from the data versioning manager 917, the default optionally being to retrieve the latest changes; however, a prior version may be specified within the data request. The changed document, for example the annotation document, is inserted into, e.g., a data versioning manager 917, to accomplish version control. There are several appropriate varieties of commercially available version control software. A version control program

generally compares the revised document against the prior copy, makes a list of the changes, and associates a new revision with those changes. Optionally, upon changing or updating a document, a change notification 923 may be initiated for use by other processes.

The object API 319 may provide for services, e.g., to insert, update, delete, and/or retrieve binary data (such as for images). Such binary data advantageously may be managed by the data versioning manager 917, and optionally a new revision may be created on update. The binary data type may be stored advantageously with the binary data when inserted or updated, e.g., in image data files 907. The various standard image types (e.g., JPEG, GIF, TIFF, etc.) optionally may be known to the system as predefined data types. Optionally if an image is requested in a different known format than it is stored, conversion to the requested format may be performed, e.g., during retrieval; the may be done advantageously by one or more format converters 925.

One of the other aspects of the optional version control system is that it is possible to label a particular version as having a given name. The label then readily allows the system to associate a version of the annotation data with a version of the document data. (The annotations and the document may be changing at different rates.) One or more embodiments of the present invention provide the ability to create an associated name with a revision, and the version control allows the system to then associate the various into version streams that are changing at different rates.

The document being edited and/or marked up, e.g., a license agreement, may be changing at a very rapid rate initially but then those versions may slow down as the license matures. Conversely, the annotations may start to grow rapidly or there may be a period when a company

is working with a particular sub-licensing arrangement where those changes are occurring rapidly as well. The XML document received from the data analyzer is then fit into the data versioning manager, 917. Relevant information optionally is stored into a repository 905 and that reflects one part of the life cycle of that document.

The optional XML repository 905 provides the data versioning manager's log file storage, for change records. It may use a form of data compression that is typically used in version control systems where storing the changes that have happened from one revision to the next of the document.

When XML data is inserted or updated in an XML document, it may be parsed by an XML parser. Optionally, a configuration data section 919 may be used to identify document structure. An index schema 913, for example, may be used to identify XML tags that the XML parser 308 uses to break up the document into major sections; and a separate index may be generated, e.g. by an index generator 915, and may be maintained for each such section. During parsing, the various elements of the XML data stream may be identified. Their contents further may be parsed to extract the individual words within each element. These extracted words may be compared against a table of unimportant words. If not matched in the table, the word, together with the unique (fully qualified) XML document name, plus its new revision number, if any, are may be stored in an index SQL database 903. Each entry (e.g. row) in the table may be identified (via e.g., primary key) by the word, the document name, and the revision, or in any other appropriate way. This table may contain a separate field (e.g., column) for each section of the XML document, which may contain a count of the number of times (e.g. frequency) the word appears within that section. This realization may enable an index searcher 911 to place the most likely candidates at the beginning of the search results. Other realizations are possible, and will

be appreciated by one of skill in the art. The data server optionally includes a thesaurus 909, which may reference and/or manage a table of synonyms to be used, inter alia, in broadening the field of search. Thesaurus 909 may maintain relevant data in any appropriate form, such as thesaurus SQL data 901.

When XML or binary data or other data is inserted, updated, or deleted in a document, a change notification 923 event optionally may be generated. This may be broadcast to registered listeners (typically one or more data managers). The change notification event may advantageously provide underlying support for replication, and/or may be used for notifying a user of modifications to a document that they may be reviewing.

Reference is made to Figure 10, a block diagram of one embodiment of an architecture for the data analyzer 203. The data analyzer communicates via the data manager to one or more data servers on the backend of the system. From a user's perspective, one optionally is interacting with the data analyzer 203 via a user interface, e.g., looking at a directory and/or search view 1001 in order to locate, edit, or annotate a document. For example, a user interface may present a directory as a navigable tree, allowing them to see one or more data managers in connection with respective data servers. Optionally, a data manager may then be responsible for presenting a relevant part of the navigable tree. Other means of displaying documents are equally appropriate, e.g., the data analyzer may show a list of the patents by year issued, by classification, etc. An interface is provided so that the user can identify a document they want to work with. For example, a user may do a search, e.g. for all documents containing the term "cup", and be presented with a list of search results, in order to access documents.

In any event, the user identifies the document of interest, and a request for the document is sent by the data analyzer to the version manager 1003, e.g., for the latest version. In some cases, especially if the user is interested in looking at historical changes, then they may want to obtain a prior version of that document. The version may be important if there is an association between the version of annotation data and the version of the document data. Optionally, the system retrieves multiple versions, e.g., to illustrate a particular moment in the history of that document. The “XML Request” may be sent as a data manager communication 1021 to a document manager 1005, and the requested XML document may be returned. The document manager 1005 reads and then stores that document into the document workspace 1007 according to one or more embodiments of the present invention.

In the document worked on by the data analyzer 203, the annotation data preferably is already merged with the original document data. The merged document may be optionally stored into a document workspace 1007. The purpose of the document workspace 1007 is so that a user may remotely work on a document, such as on a notebook computer not connected to a network. The data may be local in the workspace. Hence, if there is a disconnect from the original data sources, it is irrelevant in terms of working with the document.

When the remote user finishes edits or annotations on those one or more documents, they may then check those documents back into the data server through the data analyzer and the data manager. When a document is to be worked with, it may be extracted from the document workspace 1007. Optionally, the document to be worked on is broken down into elements, if any. The document workspace 1007 may be, for example, a file in a directory or a set of directories on a disk. To break the document into elements, according to one or more embodiments of the present invention, the document is extracted from the document workspace,

and fed into the XML parser 308 within a document object model 1011. The document object model 1011 may be, e.g., standard binary representation or object representation of XML. An annotation schema 1017 may be referred to within either the XML parser 308 and/or an XML generator 1013, for example in the process of conversion to and from the object model 1019.

Once the XML data is broken down into an internal object representation, it is possible to look at an individual element, and determine, e.g. what is the content of that element in terms of text, name of the tag, text of the entity, tag name for the entity, and/or parent entity for it.

The merged or annotated document advantageously may be provided in XML format. The XML document is a structure with a balanced mark-up tag; each tag specifies a start and end of the section, and inside a section there may be nested one or more start/end of another section one. Each one of these start/end blocks may be a node. Each entity within that becomes a sub-node of a tree, creating an in-memory representation of the document tree that can be traversed to the parent node, child node, siblings, etc. The XML object model 1019 then contains the document data and child nodes for each one of the paragraphs or items that have been annotated. An annotation node contains the annotation data. It contains the type of a mark-up, e.g., “conformance”, link to another document; textual node, etc.

Optionally, the user is provided one or more views in order to assist with analysis of the document. Each of the different views 113, 115, 117, 119 works with the merged document, according to one or more embodiments of the present invention. The views 113, 115, 117, 119 may determine document format, e.g., by a reference to a document Schema 1009. The user selects the function he wishes to perform on a document, e.g., view a report of the document

113; view a map of the document, links and mark-ups 115; view annotations 117; and/or edit the document 119.

Reference is made to both Figures 4 and 10. The editing view 119, and editing window 201 illustrate one example of editing an intellectual property document, here, the license. The editing view examines the tree structure of the document or a nesting of levels within a document. Here, there are articles at the outermost levels, sections within that, and perhaps each section has clauses with sub-clauses. They may be broken down separately. In this case, there is an article at the outer-most level, which is at the same level as the preamble, nested within that article 1, "Grant of Licenses", there is article 1.1 "Trade Secret License". The entire text for "Trade Secret Licenses" may be contained within one node in the document object model. The "Trade Secret License" tag may be contained elsewhere within that node and embedded therein; "1.1 Trade Secret License" is a node, the child node of that is the text of the paragraph, etc. At another child node of Section 1.1, there is provided a conformance mark-up (displayed as illustrated in Figure 5 in the mark-up window 501). The data and/or attributes within the conformance mark-up would indicate that there is a possible conflict, together with the contents of the text within that child node or within a further child node thereof. This mark-up information for the illustrated conformance item may be associated with Section 1.1 as a separate node.

With regard to the illustrated Section 1.1, the user may select the node indicated as selected, e.g., by a frame 405, e.g. by a double click or click inside the frame 405. The view may change to an editing view such as a frame with scroll bars. The user may modify the content of the selected information. The system automatically updates the original document information.

Reference is again made to both Figures 5 and 10, illustrating an example mark-up window 501 and annotation view 117 within one or more embodiments of a data analyzer architecture. The annotation view 117 and mark-up window 501 display the mark-up associated with document, and/or documents to which the document has been linked. For example, the systems traverse the object-model tree in memory, locate annotations that exist within that tree, and locate the particular corresponding annotation(s) for items at a given level within the tree. It is also aware of the document object model in this example.

Reference is again made to both Figures 8 and 10, illustrating the map view 115 together with the map window 801. The map view 115 reviews each of the nodes within the selected document and displays them for example, within a tree or a map format. In the present example, it displays a tree of boxes representing nodes within the document, nodes of other documents connected from the selected document, and/or annotations associated with the document; and lines connecting the boxes together, representing links from the document (or nodes therein) to other documents (or nodes therein) and/or associated annotations. It is working off of the object model in this example.

Reference is again made to both Figures 7 and 10. The report view 113 and report window 701 look at different elements of the document, nodes, and annotations and pull them together into a textual representation and/or summary.

Advantageously, each of the views 113, 115, 117, 119 is provided as a plug-in to the system architecture, or similar fashion, to enable views to be added, omitted, supplemented and/or combined. Other views may be provided, and the examples herein are provided merely for illustration of the underlying principals. Further, although it is advantageous for the views

and/or data analyzer to work on one document as a whole, the document could be provided in multiple parts and/or with separate annotations.

Reference is now made to Figure 11 showing an example embodiment of an optional data flow for splitting an annotated document. In this conceptual illustration, the annotation data stream 1123 is separate from the document data stream 1121, and the annotations and documents are stored separately. The annotated document 1127 is received by the annotation split logic 1111 and is broken apart into document output data stream 1121 and annotation output data stream 1123, e.g., for storage, e.g., in an XML document repository 1101 and an XML annotation repository 1103. The document is parsed, e.g., by an XML parser 308, and a document mark-up schema 1115 is used to help identify nodes within the document and/or annotations. If implemented using XML, tags are associated with the document, and the tags that are associated with the annotation.

Optionally, multiple versions of the document and/or annotation are managed, e.g., by respective versioning data management systems 1105, 1107.

Preferably, any kind of annotation data, and/or any kind of document data, and/or format may be accepted. They are advantageously converted into XML, and then converted from XML into their native format.

Figure 12 further illustrates that there may be two or more input data streams 1203, 1205, retrieved from the XML document repository 1101 and XML annotation repository 1103, for a particular marked-up document, which are merged together in accordance with one or more embodiments of the present invention. At least one set of the input data streams contains document data 1203, and at least one other set of the input data streams contains the annotation

data 1205 to be applied to such document data 1203. Annotation merge logic 1201 identifies locations in the document into which to associate annotation data. If the document is XML, e.g., an XML parser 308 may utilize the document mark-up schema 1115 to identify appropriate locations.

If more than one document is embedded within a stream, the system may extract that document from multiple documents embedded within a single stream, in order to obtain a single-document stream in any event.

The document data may optionally be provided in multiple document streams. In the case of the USPTO database, data from 1976 to 2000 is stored in a formatted character mode, which is non-standard and awkward to handle. This information is stored as provided by the USPTO, in multiple files per patent. Those files contain the abstract information, information about the inventor, a brief description of the claims, drawings, etc., so there are several documents for a given patent. Optionally, all of the annotations that relate to one other document could be stored in one annotation stream, and all of the annotations relating to yet a different document optionally may be stored in a separate annotation data stream. There is no requirement that all annotations for a document come from or be stored into a single annotation file.

Annotation merge logic 1201 inputs the input data streams 1203, 1205, and creates a mark-up representation of the document data, containing, referencing or including the annotation data, whether by structure or reference for associating the annotation data with its corresponding elements within the document data.

Figure 12 further illustrates a document input data stream 1203 containing document data, and an annotation input data stream 1205 containing annotation data. The annotation

merge logic 1201 outputs the result of the merge, i.e., a marked-up output data stream 1207. The representation of marked up output data can reference the annotation information in many different ways. XML is fairly flexible and one advantageously may define the annotations at the top of the document as entities. Accordingly, one may take text as written, paste it into the XML document and then re-parse the document, to further evaluate the XML structure.

The XML element is a macro that may be cut, pasted and inserted into another section of the document by reference. Hence, one alternative according to one or more embodiments of the present invention is to take the annotation data, define each one of them as elements at the top of the file, and then simply embed a reference to that element within each of the paragraphs where it needs to be expanded. That provides the mark-up copy, and it is semantically equivalent to embedding the actual mark-up entities within the entities that they refer to in the original document.

There are several alternative ways to include the annotations, e.g., write the annotation to a separate XML file, and use an include statement to include the contents of that XML file. The concept of the different ways of expanding into the mark-up document may be realigned in different ways, whether by inclusion of an element, the macro-type element, by doing an include to pull it in from another document, or by expanding out the XML code for such representation further containing, referencing or including the annotation data.

There are a number of alternative ways in which the data may be provided. The data stream could be, for example, a named pipe, data from a firewall, data from a disk, or data from a database, etc.

According to one or more alternative embodiments of the invention, the document data and/or the annotation data are stored in multiple data servers, and may be accessed via one or more data managers. For example, data might be distributed among servers physically located, e.g., at a global headquarters of an information service, a corporate headquarters of a company, of a small law office, and/or a personal computer.

According to one or more alternative embodiments, the document data and/or annotation data and/or marked up document are provided as data streams. If a data stream contains image data or other binary data, one of the data streams may include data for associating the image or binary data with the annotation data and/or document data. This is useful if, for example, there are images that are associated with many of the patents, trademarks, etc.

The image/binary data stream is not necessarily distinct from the document data stream or, if appropriate, the annotation data stream. The document itself may contain a reference to an image, and/or the annotation itself may contain the reference. In one or more embodiments of the present invention, on the other hand, the image/binary data stream might or might not be distinct from the document data or the annotation data.

According to one or more embodiments of the present invention, annotation data may contain an association of an external data stream of, e.g. document data. The annotation data may have an association to external data, e.g., a hyperlink to a URL web page, a fully-qualified file name on a network server, the document, a name of a program, a name of a command string that can be executed through a command shell to start, e.g., a computer aided design (CAD) system with a particular CAD file, etc.

According to one or more embodiments of the present invention, associations may be formed between the version of an annotation with a version of the document. Preferably, one or more of the input data streams is from a versioning system, where there is provided a version control system, with multiple versions of a document and/or annotation. The system and/or user selects one of those document versions and/or annotation data, from the versioning system. Where both document data and annotation data are provided from a versioning system, there may be one or multiple versioning systems.

Marked up input data streams may contain annotation text, or may be related to a stream that contains annotation text. According to one or more embodiments of the present invention, a marked up document may be received as an input data stream or marked up document coming in to an input data stream. Annotation data may be included that is associated with, embedded in, or connected with the input data stream. The input data streams may include, inter alia, annotation data, and/or a marked up document representation. The system is capable of parsing such marked up document representation. The system may extract from such marked up document representation the annotation data which may be placed into one or more output data streams. The annotations are optionally stripped out, and made separate and distinct from the marked up data stream.

The system can review the marked up document, and may extract the relationship between the annotation data and the elements of the document.

According to one or more embodiments of the present invention, there is provided a user interface. When the user selects a different kind of annotation or when the content of the

annotation changes, for example, the user may dynamically change how a particular user interface displays the information that it is working with.

Depending on the type of the annotation, e.g. a conformance test, one or more parts of the user interface may display themselves differently than for history of the document. Consider that something is displayed in a user interface window. The user selects one of several different annotations that they want to work with. The screen displays the information they are working, as it changes, in one form or another.

Figure 13 illustrates an example of an annotated XML document 1301, according to one or more embodiments of the present invention. The annotated XML document 1301 includes one or more document elements 1303 embedded therein or otherwise associated therewith, with document data. One or more annotations 1305 are embedded or otherwise associated therewith. The annotation 1305 includes one or more annotation elements 1307, which reference data, a document, an external data source, etc. The annotation element 1307 may have a link 1309 to zero, one or more external data streams. In this example, a link 1309 is provided to data streams including a document element data 1313 within another document 1311; a URL 1315, e.g., “http://www.www.xyz/page”; and other external data source 1317, e.g., an executable shell script, image file, diagram, text file, document, or other file (voice, audio, video, binary, etc.)

Reference is now made to Figures 14A-B, illustrating an example flow chart for merging document data together with annotation data to produce a marked-up representation of the document. At step 1401, the user selects a document to be marked-up. At step 1403, the system determines whether the currently located document is the correct document for marking-up. If not, at step 1405, the system searches for the correct document. Once the correct document is

obtained, at block 1407 the system determines whether the current version is the correct version. If not, the system searches for the current version of the document at block 1409. Once the correct version of the current document is obtained, at block 1411, the system determines whether there is any annotation data for the selected document, for the particular user. If the current annotation data is not the correct annotation data, at block 1413, the system continues to search for the annotation data corresponding to the selected document, block 1417. At block 1415, if the current version of the annotation data is not the correct version of annotation data, then at block 1419 the system continues to search for the correct version of the annotation data.

At block 1421, the system has the correct version of both the selected document and the annotation data, and the system proceeds to place the document data into a mark-up representation of the document. At block 1423, the system loops to check for additional items of annotation data. For another item of annotation data, at block 1425 the system locates the corresponding element within the mark-up representation of the document, and at block 1429, the system, associates the annotation data with the corresponding element of the document. When there are no further items of annotation data, at block 1427 the system provides the user with a marked-up representation of the document. Processing ends at block 1431.

Reference is now made to Figure 15, illustrating one example of splitting of a marked-up representation of a document into annotation data and document data. At block 1501, the system obtains a marked-up representation of the document. In blocks 1503, 1507, 1511, 1515 and 1517, the system loops to obtain each element in the marked-up representation of the document, determine the annotation(s) in the element, and split out and store the annotations. In blocks 1505 and 1509, the system separately stores the document data and annotation data. Hence, in block 1503, the system, determines whether there is another element in the document. If so, the

system obtains the next element in the marked-up representation of the document at block 1507. At block 1511, the system checks whether the element includes one or more annotations. If so, the system stores the annotation(s) in the annotation data at block 1515. At block 1517, the system stores the element in the document data. The system loops back to block 1503 for the next element in the document. Once done processing elements in the document, the system stores the document data, as a new version, for this user, at block 1505; and stores the annotation data, as a new version, for this user, at block 1509. At block 1513, the system returns from processing.

Figure 16 is a linked diagram illustrating an example of linked, annotated intellectual property documents and data, according to one or more embodiments of the present invention. Here, one or more users has linked together several related intellectual property documents, in this example including a text document 1601 (titled “Power Projects”), technical description documents 1603, 1605 (titled “Jet Engine” and “Turbine Engine”), a patent infringement analysis 1607, and several patents 1609a-h. In this example, associations between two documents are illustrated by links 1613. A document may be linked one way or both ways. A link may be to/from the document generally, or a specific location in the document. Each link may include an annotation 1611. Preferably, the annotation includes any user comments, user-supplied text, other user-supplied digital data, user-defined attributes (e.g., company’s patent, competitor’s patent, project name), history, etc. In the present example, a user could select the “Power Projects”, view the links and embedded annotations regarding the “Jet Engine” and “Turbine Engine” documents. The user could select one or more of the links to linked other intellectual property documents. The process continues throughout the chain of linked documents. The user optionally may select yet another intellectual property document and create a link with optional

annotation. An intellectual property document may be multiply-linked, and may link to itself if desired.

Figure 17 is a linked diagram illustrating another example of annotated intellectual property documents and data, according to one or more embodiments of the present invention. The subject of this example is a license 1701 including multiple terms 1713. The license generally is linked both ways to a related product document 1715. The license includes annotations with internal notes 1705, 1707 on two terms; an annotation of a term with multiple versions of proposed changes to a license term 1709; an annotation relating to two terms with a digitized voice recording of a negotiation 1711; and a link both ways to a related patent, trademark or other intellectual property document 1703, with annotations 1611.

Figures 18 A-B is an example flow chart illustrating an interaction with the user to obtain annotations and links for an intellectual property document, in accordance with one or more embodiments of the present invention. At step 1801, the document to be marked-up is provided to the user, for example via a display. The document may have been previously obtained, for example via a search, browse, or other retrieval component, tool or function. At step 1802, the system interacts with the user to determine a portion of the document to be marked-up. The document may have been previously divided into sections and/or subsections, for example, that are candidates for marking up. Alternatively, the user may, e.g., perform a click-and-select function to selected a portion. At step 1804, the system optionally indicates the determined portion, for example, by highlighting the portion, via a pop-up-window, via special color, etc. At step 1806, the system interacts with the user to obtain a mark-up for this portion of the document. For example, the system may provided a pull-down menu, a pop-up window, a particular font, etc. The permissible contents of mark-up to be applied may be customized by an

administrative user, may be free-form, and/or may have a check-list of pre-defined elements, etc. According to one or more embodiments of the present invention, the user may select and/or enter the mark-up information. At step 1808, the system determines whether the mark-up is to include one or more references to an intellectual property document. If so, then at step 1810, the system provides that the user can locate and/or link from the present document to the intellectual property document. In the present example, the system provides a search and/or browse tool to locate the document. At step 1814, the system interacts with the user to indicate a selected portion of the document to be linked to. The selected portion may be some or all of the current document, and/or another document. At step 1818, the system saves a reference, e.g., a link, pointer, identifier, for the other document and any selected portion, together with the associated mark-up. At step 1812, the system saves the mark-up, together with any optional reference to another document and/or the indicated portion thereof, into, for example, temporary storage. At block 1816, the system checks whether there are any further mark-ups to be applied to the current document, and if so, loops back to step 1802.

If there are no further mark-ups and if the document and mark-ups are to be saved, then at block 1820, the system determines whether the marked-up document was edited and/or was editable. If so, the document is stored at step 1822. At step 1824, the system determines whether there is one or more saved mark-ups to be applied to the document. If not, then the system exits. If there are mark-ups, then at step 1826, the system determines whether the mark-ups are stored separately from the document. If not, then at step 1828 the system stores the saved mark-ups together with the document. Otherwise, at step 1830, the system stores the saved mark-ups separately from the document, and at step 1832 stores data representative of the mark-up locations within the document. The function then exits processing.

Figure 19 is a flow chart illustrating one example of traversing from intellectual property document to intellectual property document, via links associated with the document and/or sections thereof, optionally having annotations. At step 1901, the system obtains the document, and displays the document together with annotations (or indications thereof). At step 1903, the system loops for the user to select an annotation and/or section of the document associated with a link. At step 1905, the system displays the annotation information, if any. At step 1907, the system determines whether the annotation (or selected section) includes or is associated with one or more links. If not, the system loops back to step 1903. If there is at least one link associated with the annotation (or selected section), step 1907, then the system loops at step 1909 until the user selects a link. When the user selects a link, then at step 1911, the system determines the location of the linked document (or section thereof) via reference information, for example, stored or associated with the annotation, obtains the linked document (or section thereof), and displays the just-obtained document, optionally together with any annotation indications. The system then loops back to step 1903, enabling the user thereby to continue to traverse the related linked documents.

Reference is now made to Figure 20, illustrating an example architecture for use in connection with one or more embodiments of the present invention. In the present example, a computer 2001 hosts one or more annotations components 2003 and one or more linkages components 2005. The annotations component has one or more of the following: a component to apply an annotation 2013 to a document; a component to edit an annotation 2015; and a component for document and/or section selection 2017. The apply annotation component 2013 interacts with the user to create an annotation, e.g., using menus, free form text, cut-and-paste of text, web pages and/or hyper links; and to apply that annotation to the document (or to the

selected section of the document). The annotation may be applied, e.g., by inserting the annotation into the document, by saving the annotation separately in an annotations database 2011 and inserting a reference to the annotation into the document, and/or by saving metadata associating the reference and the document (or selected section thereof), etc. The edit annotation component 2015 interacts with the user to edit an existing annotation, e.g., using menus, free form text, cut-and-paste, etc., and optionally to save the edited annotation. The edited annotation may be saved, e.g., by saving the edited annotation with the document, by saving the edited annotation separately and optionally updating a reference to the annotation into the document, and/or by updating metadata associating the reference and the document (or selected section thereof), etc. The document and/or section selection component 2017 interacts with the user to determine a portion, portions or the entirety of the document to be associated with the annotation.

The linkages component(s) 2005 include one or more of: a component to establish, indicate and/or remove one or more links 2019, a component to allow the user to traverse one or more links 2021, and a component for document and/or section selection. The document and/or section selection component 2023 interacts with the user to determine a portion, portions or the entirety of one or more documents to be associated with a link. A link may be between one or more documents or sections thereof. A document may be linked back to itself or a section therein. The component to establish, indicate and/or remove a link 2019 interacts with the user to determine the document and/or section to link from, and the document and/or section to link to. The link may be established or indicated, e.g., by inserting a link (e.g., reference, pointer, etc.) into the document, by saving the links separately in a links database 2009 and inserting a reference to the link into the document, and/or by saving metadata associating the link and the

document (or selected section thereof), etc. Optionally, links and annotations are stored in association. Optionally, links are stored within the associated annotations, or vice versa. The component to traverse links 2021 determines one or more links, if any, associated with a selected document and/or selected portions thereof, optionally one or more annotations associated therewith, and optionally the document title or description at the node of the link. Further, the links component 2021 interacts with the user to determine which link to traverse; to obtain the link (pointer, reference, etc.) to the linked document; and to retrieve the linked document and provide to the user. With the retrieved document, the user may traverse further links therefrom. According to one or more embodiments of the present invention, one or more users 2027 are local communicating with the computer 2001, and/or are connected over a network, e.g., the Internet 1005. In the illustrated example, the documents database 2007, links database 2009, and annotations database 2011 are local to the computer 2001; a further documents database 2025 is accessed via the Internet 1005.

Reference is now made to Figures 24 to 26, illustrating one or more aspects of the present invention. Types of attributes assigned to intellectual property documents, and/or the attributes themselves, may be used in connection with searching, accessing, analyzing, and/or reporting, for example. In at least some instances, attributes and attribute types are different from pre-existing field contents or fields (e.g., inventor, title, serial number, etc.) in the intellectual property document.

Attributes are conveniently assigned when there is an intake of one or more intellectual property documents to the intellectual property documents database. This may occur at various points in time, for example, when the intellectual property document is initially available (e.g., upon initial conception, creation or upon filing), and/or first designated for inclusion, and/or

when adding legacy intellectual property documents. An intellectual property document that already has attributes assigned, may be subjected to further assignment, deletion, and/or revision of attributes.

According to one or more embodiments of the present invention, attributes associated with an intellectual property document are stored apart from the intellectual property document. For example, attributes may be stored as metadata. Optionally, attributes are stored in conjunction with other annotations associated with the intellectual property documents. A unique identifier is used to locate the related metadata. A serial number, issue number, other unique identifier, or portion thereof, optionally in combination with, e.g., a country code or intellectual property type indicator, may be used to provide a unique identifier.

A type of attribute may reflect that the intellectual property documents are product-centric and/or services-centric. For many companies or concerns, everything they sell is tied to a product or a service. Other traits or information may be used as an attribute type. Once a user is able to group intellectual property documents, and label the intellectual property documents as being related to one or more attributes and attribute types, then the attributes and attribute types may be used for various applications, e.g., searching, generating reports, etc. Hence, attribute types may include one or more of: a product, a service, an actor (a person or entity who performed an action), a user, the current owner (as provided by the user, e.g., not derived by the system from inherent fields or other data), an indication that the patent is the company's or a competitor's, a project name, and/or an indication of level of conformance.

Referring once more to the example of product/services as a type of attribute, assume that a user has determined one or more particular products or services to be associated as an attribute

with multiple intellectual property documents. A user can identify the patent(s) and/or other intellectual property associated with a particular product or service by providing information identifying the product or service. Conversely, by identifying a particular patent, for example, a user may determine the products that are relevant to the patent.

Consider for example, that a user wishes to analyze a patent from the perspective that the user owns the patent. In this example, the user specifies the patent, and determines that the patent relates to a particular component. The user may then request a report to see all patents associated with the particular component. Assume further that the attribute type of component is a sub-type of the product attribute type. Hence, the particular component relates to a particular product. The user may then request a report to see all patents associated with the particular product.

By providing one or more attributes and/or attribute types by, e.g., pull-down menu, tree structure, or check-list, a user may easily determine the attribute or attribute type that is desired. For a user with a product- or service-oriented viewpoint, the attribute types may relate to products and/or services. The attributes and/or attribute types may be pre-defined by the system, and/or may be user-customized, e.g., by an administrative user.

An intellectual property document may be associated with one or more attribute types, and an attribute type associated therewith may be assigned one or more attributes. For example, a patent for a chemical dye may have a product type of “clothing, food, and hair care.” A query for intellectual property documents related to the product “clothing” would return results including the patent for the chemical dye.

According to one or more embodiments of the present invention, a company or user may customize the attributes and/or attribute types. According to one or more embodiments of the present invention, one or more attributes and/or attribute types may be locked so as to be unchangeable except by an appropriately authorized user. An attribute optionally permits addition of free-form information by the user, e.g., text, URL to a file, a reference to another intellectual property document, an image file, a video file, an audio file, or a file made using another computer software application.

Optionally, one or more attributes to be associated with a particular intellectual property document may be determined through, e.g., a scanned or typed SKU, model number and/or a manufacturer batch number.

Optionally, information regarding attributes and attribute types can be exported or otherwise provided for use in connection with another entity that has a separate collection of intellectual property documents. Hence, according to one or more embodiments of the present invention, a company may send its attribute data to an outside IP boutique for use in connection with the IP boutique's intellectual property documents.

A further optional feature provides for preliminarily determining a grouping of intellectual property documents to be associated with attributes. A user selects a grouping of intellectual property documents to constitute a project. The documents to be included in the grouping may be specified in any appropriate manner, e.g., as search results, specified by registration number, a patent family search, etc.

The project can be subdivided to provide another project grouping. The intellectual property documents within the project (whether the original or the sub-project) can then be

worked on. This is useful, for example, where large number of intellectual property documents need to have attributes assigned. For example, a user could request all patents owned by the General Electric Company (GE), and subdivide those into multiple projects for further action by other users in associating appropriate attributes therewith.

Optionally, attributes may be assigned to one or more intellectual property documents within a project grouping, individually and/or as a group.

As a further option, an intellectual property document may be included in more than one project.

Optionally, an ownership request performs multiple database checks to more accurately determine ownership. With respect to patents, for example, a patent ownership search would locate patents issued to GE (as listed in the patent data), plus patents currently assigned in to GE (as listed in assignment data), minus patents assigned away from GE (as listed in assignment data).

Optionally, the grouping takes into account intellectual property that has lapsed, e.g., expired, canceled, for failure to pay maintenance or annuity fees. For example, the user can specify that the grouping should omit, include, and/or be limited to lapsed intellectual property.

Optionally, the intellectual property documents within the project are sorted to provide for coherent presentation to a user. Documents may be organized in any appropriate manner, for example, by intellectual property type (e.g., patent, trademark, etc), by fields within the documents (e.g., class/subclass, XML tags) and/or by previously assigned attributes. For example, the project grouping of all patents owned by GE could be sorted by class, and then

subdivided (e.g., along class lines) into further projects. The sorting and/or further subdivision of intellectual property documents within a project may be manual, automatic, or semi-manual.

A project grouping of intellectual property documents may be provided for further action by a user. Optionally, a user can check in and check out from projects. Optionally, one or more intellectual property documents in a project may be re-assigned to a different project.

One or more embodiments of the present invention provide for utilizing the attributes as a filter. For example, the user could filter the intellectual property documents and view all documents that have a specified attribute or attribute(s). Filtration utilizing attributes could be used, for example, with searching, retrieving, reporting and/or viewing. Optionally, filtration could utilize one or a combination of: (i) attribute content; (ii) type of attribute; (iii) sub-types of attribute types; (iv) content of a field in the intellectual property document; (v) type of a field in the intellectual property document; and/or (vi) information derived from one or more of the foregoing, e.g., current owner of record. Optionally, where attributes are stored as metadata, the metadata can be searched to determine the intellectual property document(s) having specified attributes.

The determination of intellectual property documents included in a report generated using attributes and/or attribute types as a filter can be very flexible. One or more of the attributes and/or attribute types associated with the subjects of a report may be included as or the basis for information in the report. For example, a report could count the number of trademarks owned by a company for each of several products or service lines.

Reference is now made to Figures 24-26, providing examples of assigning attributes to intellectual property documents, grouping documents into a project, and using attributes for

filtering. Figure 24 is an example flowchart for determining attributes, step 2401, to be associated with, for example, intellectual property documents or files. At step 2403, one or more intellectual property documents are selected to be assigned an attribute. At step 2405, the system interacts with the user (or automatically) to select an attribute type. At step 2407, the system interacts with the user to select one or more attributes (based on the attribute type) to be assigned to the document. At step 2409, the system associates the attribute type and the attribute with the document. At step 2411, the system determines whether it is desired to assign another attribute for this same attribute type to this document, and if so, loops back to step 2407. At step 2413, the system determines whether an attribute for another attribute type is to be assigned to the current document, and if so, loops back to step 2405. At step 2415, the system determines whether there is another document to which to assign attributes. If so, at step 2417, the system gets the next intellectual property document to be assigned an attribute, and loops back to step 2403. If the system is done assigning attributes to documents, at step 2419, the system stores the attribute type(s) and the attribute(s) that were associated with the document. At step 2421, processing for this flow ends.

In accordance with one or more embodiments of the present invention, the attributes may be provided in a hierarchy structure with attributes, sub-attributes, sub-sub-attributes and so forth to create a tree structure. An intellectual property document or file may be associated or tagged with anyone or more of these attributes and sub- or sub-sub-attributes. At the time of association or tagging, the intellectual property document or file is simultaneously or automatically associated or tagged with all of the attributes that are at a level higher than the particular attribute or attributes with which the intellectual property document or file is associated or tagged. In other words, the intellectual property document or file adopts or inherits the tag or associated

profile of the attribute or attributes with which it is associated or tagged. In this manner, all of the attributes that otherwise would have been selected in a step-by-step manner may be assigned simultaneously and automatically by tagging the intellectual property document or file with not only the selected attribute(s), but all of the other attributes that are at a level higher than the selected attributes.

Optionally, intellectual property documents to be assigned attributes are grouped into one or more projects together prior to the assigning process. The assigning process can then intake the intellectual property documents in a particular project. Grouping of large amounts of intellectual property documents can be assisted by providing an ability to sort and/or subdivide. It is understood that the grouping of intellectual property documents or files is also effected by tagging.

Figure 25 is an example flowchart for grouping intellectual property documents into a project, step 2501. At step 2503, a grouping of IP documents to be ingested is selected, for example, by utilizing search results or documents in a workspace. At step 2505, the system interacts with the user to determine whether the grouping needs to be sorted, and if so, at step 2507, the system sorts the grouping of documents. At step 2509, the system interacts with the user to determine whether the grouping needs to be divided, and if so, at step 2511, the system determines documents to be in the sub-groupings, for example, by selecting particular documents or placing a portion of documents into a folder. At step 2513, the system determines whether the sub-grouping itself needs to be sorted and/or further subdivided. If the sub-grouping needs further sorting or subdivision, the system loops back to step 2505. Sub-grouping can be re-iterated. If there is no further grouping, sorting, and/or dividing, then processing for this flow ends at step 2515. It is to be understood that each grouping may also effected by tagging.

Once attributes are assigned to intellectual property documents, the attributes can be used to filter, e.g., for searching and/or reporting. Figure 26 is an example flow chart for using attributes to filter through intellectual property documents, 2601. At step 2603, the system interacts with the user to determine the attribute(s) to be used as a filter. At step 2605, the system searches the meta-data for the specified attributes, where the meta data has the document identifier (or other reference to the document). At step 2607, the system determines whether the search of the meta data is done, and if not, loops back to step 2605. At step 2609, the system displays information representative of the search results, e.g., document identifiers (serial numbers, etc.). Optionally, the system obtains further information concerning the document (e.g., title, mark, owner). Therefore, at step 2611, the system determines whether it should retrieve data for the intellectual property document. If so, at step 2613, the system retrieves the intellectual property document data for the search results. At step 2615, processing for this flow ends.

The following provides an example use of one or more embodiments of the present invention. Initially, ABC Corp. decides to intake the intellectual property portfolio of a company it acquired, XYZ Corp. In this example, the XYZ Corp. intellectual property assets have not yet been assigned attributes. The ABC Corp. user performs a search for all patents and trademarks with XYZ Corp. as the current owner of record. This provides a list (for example) of 2,000 patents and 200 trademarks. The user sorts the list, first by type of intellectual property; the user then sorts the patents by class/sub-class, and the trademarks by International Class. The user then selects, drags and drops, into various projects, the sorted intellectual property. (The projects can be worked on by other users, and provide a convenient way to subdivide large numbers of documents.) By working through documents within a project, a user then may assign attributes

to the intellectual property therein. For example, consider that one project has patents for copier technology. The user selects one or more patents in the project and chooses to assign attributes. The system prompts the user to provide an attribute type, e.g., product. Attribute types may be customized to include, in this example, Company, Division and Technology Type. Acceptable attributes for the selected attribute type(s) are displayed to the user; optionally, the user may type in text. In our example, the user selects:

| <u>Attribute Type</u> | <u>Attribute</u> |
|-----------------------|---------------------|
| Company | ABC Corp. |
| Division | Electrical Division |
| Technology type | copier technology |
| Product | Model 123 |

This process is eventually performed for all of the documents in the project. Other searches and/or sorting can be performed if desired to better identify documents to be grouped into projects. Further, the intake can be performed on a periodic basis, for example when patents and registrations issue. Portions of the process may be automated if preferred. Optionally, a selection of a particular product (e.g., model number, component name) as a product type will cause the higher-level attribute types to inherit the appropriate attributes. For example, a selection of “Copier Technology” causes the “company” and “division” attribute types to inherit the attributes “ABC Corp.” and “Electrical Division”, respectively. Optionally, an intellectual document may have multiple attributes for an attribute type.

Continuing with the same example, a user later wants to retrieve a list of trademarks for a particular product. A search can be performed for such trademarks by, e.g., the attribute type “product” and the attribute of a specified model number; and the search will return, as results, the

trademarks that were assigned to the specified model number. This search, filtered by an attribute, optionally may be combined with searching based on pre-existing or inherent document field contents, such as “country,” to yield, e.g., a list of trademarks on a particular product sorted alphabetically and by country. As another example on the patent side, a search for a particular product would yield a list of patents (if any) relevant to the product.

Similarly, an intake can be performed on copyrights. Optionally, the copyright information is linked to an electronic copy of the copyright document. The user can retrieve a list of ingested copyrights, e.g., for a particular product. Documents and/or electronic representations of information relating to other intellectual property are optionally included in the system. For example, trade secret-related information may be reflected in documents specifying treatment of building access, etc.; technical data rights may be reflected in licenses; contracts may be reflected in licenses, non-disclosure agreements, memoranda of understanding, joint development agreements, joint venture agreements, etc.

Optionally, one or more embodiments of the present invention include an invention disclosure module, with optional full life cycle tracking. The optionally invention disclosure module automatically tags an “invention” directed to a particular product, beginning e.g., with the invention disclosure. Upon further creation of additional documents, e.g., draft patent application, white paper, patent, licenses, etc. the user is prompted to indicate the invention to which they pertain. A search can then be performed that results in a list of documents pertaining to a particular invention. Preferably, the documents are sorted in chronological order to reflect progress during the invention life cycle. An “invention” optionally includes multiple applications.

The use of the optional multiple-layer attributes permits a user to search, retrieve, or otherwise access one or more intellectual property documents via one or more attributes within the multiple layers. Similarly, by accessing the document(s), the user can easily determine each attribute within the multiple layers. For example, the user accessing a user interface patent assigned the attribute "Model 123" for the attribute type "Product" can determine the remaining attributes, e.g., "copier technology", "Electrical Division", and "ABC Corp." According to one or more embodiments of the present invention, an intellectual property document may be assigned to one or more sets of attributes. For example, the just-mentioned user interface patent may optionally be assigned an additional attribute "Model 456" which inherits the attributes "telephone technology", "Communications Division" and "ABC Corp." for the respective attribute types "Technology type," "Division," and "Company." A list of intellectual property corresponding to one or more specified attributes may be generated.

Figure 21 is an illustration of a computer 58 used for implementing the computer processing in accordance with a computer-implemented embodiment of the present invention. The procedures described above may be presented in terms of program procedures executed on, for example, a computer or network of computers.

Viewed externally in Figure 21, computer 48 has a central processing unit (CPU) 68 having disk drives 69, 70. Disk drives 69, 70 are merely symbolic of a number of disk drives that might be accommodated by computer 58. Typically, these might be one or more of the following: a floppy disk drive 69, a hard disk drive (not shown), and a CD ROM or digital video disk, as indicated by the slot at 70. The number and type of drives varies, typically with different computer configurations. Disk drives 69, 70 are, in fact, options, and for space considerations,

may be omitted from the computer system used in conjunction with the processes described herein.

Computer 58 also has a display 71 upon which information may be displayed. The display is optional for the computer used in conjunction with the system described herein. A keyboard 72 and/or a pointing device 73, such as a mouse 73, may be provided as input devices to interface with central processing unit 68. To increase input efficiency, keyboard 72 may be supplemented or replaced with a scanner, card reader, or other data input device. The pointing device 73 may be a mouse, touch pad control device, track ball device, or any other type of pointing device.

Alternatively, referring to Figure 23, computer 58 may also include a CD ROM reader 95 and CD recorder 96, which are interconnected by a bus 97 along with other peripheral devices 98 supported by the bus structure and protocol. Bus 97 serves as the main information highway interconnecting other components of the computer. It is connected via an interface 99 to the computer 58.

Figure 22 illustrates a block diagram of the internal hardware of the computer of Figure 21. CPU 75 is the central processing unit of the system, performing calculations and logic operations required to execute a program. Read only memory (ROM) 76 and random access memory (RAM) 77 constitute the main memory of the computer.

Disk controller 78 interfaces one or more disk drives to the system bus 74. These disk drives may be floppy disk drives such as 79, or CD ROM or DVD (digital video/versatile disk) drives, as at 80, or internal or external hard drives 81. As previously indicated these various disk drives and disk controllers are optional devices.

A display interface 82 permits information from bus 74 to be displayed on the display 83. Again, as indicated, the display 83 is an optional accessory for a central or remote computer in the communication network, as are infrared receiver 88 and transmitter 89. Communication with external devices occurs using communications port 84.

In addition to the standard components of the computer, the computer may also include an interface 85, which allows for data input through the keyboard 86 or pointing device, such as a mouse 87.

Conventional processing system architecture is more fully discussed in Computer Organization and Architecture, by William Stallings, MacMillan Publishing Co. (3d ed. 1993). Conventional processing system network design is more fully discussed in Data Network Design, by Darren L. Spohn, McGraw-Hill, Inc. (1993). Conventional data communications is more fully discussed in Data Communications Principles, by R.D. Gitlin, J.F. Hayes, and S.B. Weinstein, Plenum Press (1992), and in The Irwin Handbook of Telecommunications, by James Harry Green, Irwin Professional Publishing (2d ed. 1992). Each of the foregoing publications is incorporated herein by reference.

The foregoing detailed description includes many specific details. The inclusion of such detail is for the purpose of illustration only and should not be understood to limit the invention. In addition, features in one embodiment may be combined with features in other embodiments of the invention. Various changes may be made without departing from the scope of the invention as defined in the following claims.

As one example, the information system may include a general purpose computer, or a specially programmed special purpose computer. It may be implemented as a distributed

computer system rather than a single computer. Similarly, a communications link may be World Wide Web, a modem over a POTS line, and/or any other method of communicating between computers and/or users. Moreover, the processing could be controlled by a software program on one or more computer system or processors, or could even be partially or wholly implemented in hardware.

This invention is not limited to particular types of intellectual property. It is intended for use with any type of intellectual property, e.g., patents, trademarks, trade secrets, designs, sui generis protection, copyrights, licenses, litigations, and/or other rights. Further, various aspects of one or more embodiments of the present invention are useful with documents including those not related to intellectual property.

Further, the invention is not limited to particular protocols for communication. Any appropriate communication protocol may be used.

The report may be developed in connection with HTML display format. Although HTML is the preferred display format, it is possible to utilize alternative display formats for displaying a report and obtaining user instructions. The invention has been discussed in connection with particular examples. However, the principles apply equally to other examples and/or realizations. Naturally, the relevant data may differ, as appropriate.

Further, this invention has been discussed in certain examples as if it is made available by a provider to a single customer with a single site. The invention may be used by numerous customers, if preferred. Also, the invention may be utilized by customers with multiple sites and/or agents and/or licensee-type arrangements.

This invention has been described in connection with example data formats, for example XML and USPTO defined XML. However, the invention may be used in connection with other data formats, structured and/or unstructured, unitary and/or distributed.

The system used in connection with the invention may rely on the integration of various components including, as appropriate and/or if desired, hardware and software servers, applications software, database engines, server area networks, firewall and SSL security, production back-up systems, and/or applications interface software. The configuration may be, preferably, network-based and optionally utilizes the Internet as an exemplary primary interface with the customer for information delivery.

The system may store collected information in a database. An appropriate database may be on a standard server, for example, a small Sun Sparc or other remote location. The database is optionally an MSQl, MYSQL, mini sequel server MiniSQL, or Oracle. Information is stored in the database, and optionally stored and backed up by a back-up server, periodically or aperiodically, for example, every night along with all other data in the servers that are behind the corporate firewall into a back-up storage facility. Back-up storage facility comprises, for example, one or more tape silos that are also used to back up the entire network every night. Data security and segregation of the various customers' data is advantageously maintained. The information, for example, will eventually get stored, for example, on a platform that may, for example be UNIX-based.

The various databases may be in, for example, a UNIX format, but other standard data formats may also be used. Windows NT, for example, is used, but other standard operating

systems may also be used. Optionally, the various databases include a conversion system capable of receiving data in various standard formats.

From the user's perspective, according to some embodiments the user may access the public Internet or other suitable network and look at its specific information at any time from any location as long as it has Internet or other suitable access. For example, the user opens its standard web browser, goes to the address that is specified for its load data, and optionally fills out a user ID to log on, and a password to identify it as the specific user or the specific customer of that particular information.

Optionally, security of the networks is as tight as possible such that the data, not only customer data, but any information that is beyond the firewall is always protected against any kind of potential intrusion. The user, and, indeed, multiple users concurrently can look at the same information. Advantageously, having this system on the Internet enables users at various locations throughout the country or the world, to visit the same site at the same time and enter into a discussion or talk group as to what they are seeing, what it means, and maybe what they can do with that information.